CMSI 371-01
Computer Graphics
Spring 2016

Assignment 0308 Feedback

Outcomes that eventually cover both 2D and 3D continue to max out at **|** for now because this assignment remains in 2D.

Chris Franco

cf7 / spe861@gmail.com

*Notes while running (high-priority notes are marked with ***):*

- All of the filters run well—and a nice variety too, with a bonus of > 2 filters of each type.
- Circle gradients show up fine, though run a little long. We'll have to look at the code to see what's going on there.

Code review (refer to http://lmucs.github.io/hacking-guidelines/ for code-review abbreviations):

1. The single-pixel filters show a nice variety of color calculations, though the neighborhood pixels are more like variations on a theme: scan the neighborhood, and pick a color based on some condition. Ideally you'd have some that also perform some kind of computation involving the overall neighborhood. The `sunlight` one, for example, doesn't even need a loop—it just bases its color on the color in the upper left corner. (+*2c*, +*3c*, *4b*)
2. The circle gradients, as mentioned, calculate the gradient correctly, but here is the implementation flaw: *the circle is filled for every vertex provided to* `plotCirclePoints`*!* This is easy to see: if you take out the loop in the top-level circle functions, you'll *still* get the full circle. That is the reason for the big performance downgrade. Of course, the idea here is to genuinely take advantage of the octant that `plotCirclePoints` is being given, one vertex at a time, and make sure to fill the portion of the circle "covered" solely by that vertex. (*1a*, *2d*, *4a*)

1a — **|** …A miss for losing the benefit of having `plotCirclePoints` as a helper function.

2c (max **|**) — **|**

2d — **|** …Ditto.

3c — **+**

4a — **+** …Everything works, but…

4b — **|** …not optimally. (also covers the issues noted with the neighborhood filters)

4c — **+**

4d — **+**

4e — Descriptive messages, and even better frequency and time management. (**+**)

4f — Submitted on time. (**+**)